IN THE SPECIFICATION:

Please amend the specification as follows:

Page 1, paragraph 1:

C1

This application is a division of commonly-owned U.S. Application Serial No. 09/397,331, now US Patent No. 6298446, attorney docket number ~~21939-04692~~ 43426.47, filed on September 14, 1999, entitled "Method and System for Copyright Protection of Digital Images Transmitted Over Networks.

Page 3, paragraph 2:

C2

Other prior art techniques require a webmaster to modify images residing on a server computer in order to protect them. The webmaster is also required to modify his web pages accordingly, so as to reference the modified images. SafeMedia™ is a software product of Internet Expression, Inc. of Exton, PA that converts images from a standard format such as JPEG into ~~a SIF~~ SIF (Safe Image Format). SIF images can only be viewed with a SafeMedia Java viewer. SafeMedia embeds a host or domain name into an image, and checks that the image is located on the web site it was intended for. SafeMedia also includes enhanced system control for preventing screen capture by disabling a clipboard. Information about SafeMedia is available on the web at http://www.safemedia.com.

Page 4, paragraph 1:

C3

These techniques disable unauthorized copying of digital images from within web browsers, but they do not protect the images from ~~copying~~ being copied by an application external to the web browser. For example, they do not prevent a user from copying digital images displayed in his web browser by manes of an application running external to the web browser, such as an image editing tool, or by means of a Print Screen or other such command that serves to copy contents of a video buffer to a clipboard. Thus a Java applet that prevents unauthorized copying of digital images from within Netscape Communicator or Internet Explorer can be circumvented by a user pressing on a Print Screen button of his keyboard, or by a user copying and pasting from a window of his web browser to a window of another software application.

Page 12, paragraph 2:

C4

Typically, digital images are viewed over the Internet within web pages, such as hyper-text markup language (HTML) or extended markup language (XML) pages. Such web pages are electronic data files, stored on server computers, containing layout information for displaying text and graphics, and for running software applications such as Java applets. Typically, the data for the graphic objects, such as images, displayed within a web page is not contained within the web page file itself. Instead, the graphic objects reside elsewhere on the same server computer or other server computers, and the web page file contains ~~references,~~ references to the graphic objects. A reference to a graphic object specifies the network address of the computer containing the graphic object, such an IP address, together with the directory path (relative to a prescribed root directory) and filename for the graphic object.

Page 14, paragraph 2:

C5

The user may also attempt to copy the entire screen by pressing a "Print Screen" command key on the keyboard. Typically, this causes the contents of the video display buffer to be pasted onto the user's clipboard. The user may also attempt to save an image being displayed by running a software application outside of ~~is~~ his web browser. For example, an image editing application, such as Paint Shop Pro of Jasc Software, may have the capability of copying images from within web browsers to their own windows.

Page 15, paragraph 2:

C6

Similarly, controlling or disabling copying of displayed image data by other software applications is preferably accomplished within the Macintosh operating system by using a system extension to intervene with ToolBox functions. ~~Tool Box~~ ToolBox calls are managed by an array of pointers in a Trap Dispatch Table, each pointer pointing to appropriate program code. As described in more detail hereinbelow, the system extension can change these pointers so that they point to different program code. The different program code corresponds to patched ToolBox functions.

Page 15, paragraph 3:

A web server ~~administrator,~~ administrator ("webmaster") is responsible for configuring web server software and for managing web pages and image stored on a server computer. Typically, the administrator may wish to protect some of the images from unauthorized copying or use, and may wish to

C6 Cont

have other images unprotected, in accordance with instructions from the owners of the images. In a preferred embodiment, the present invention includes a management tool for managing protection for digital images residing on a server computer. The management tool preferably enables an administrator to select specific images to be protected from unauthorized copying or use as described hereinabove.

Page 25, paragraph 3:

C7

At step 418 the user selects one or more folders and/or web pages, from among a list of folder names and web page file names displayed by the user interface. In response, at step 420 the protection manager computer requests image information and protection status information from the server computer, for the images contained within the selected folders and/or for the images referenced within the selected web pages. At step 422 the server computer receives the request from the protection manager computer, and at step 424 the server computer sends the requested image information and protection status information to the protection ~~status~~ manager computer. As part of step 424 it may be necessary for the server computer to parse the selected web pages in order to identify the images reference therewithin. Parsing web pages is described hereinabove with reference to Figure 1.

Page 27, paragraph 2:

C8

When web pages are generated dynamically, the server computer cannot parse the web page for references to protected images until the web page is generated. However, when the server receives an incoming HTTP request to generate a web page, it sends the generated web page as an outgoing HTTP response back to the IP address of the originating HTTP request. In order to be able to modify the generated web page before sending it to the client, so as to replace references to protected images with ~~reference~~ references to substitute data, the present invention preferably re-submits the incoming HTTP request locally from the server computer to itself in order to be able to intercept the dynamically generated web page prior to its being sent to the client.

Page 35, paragraph 4:

C9

At step 822 the data returned from the operating system function is written to the clipboard and at step 824 the user pastes the data from the clipboard into a window of another software application, or ~~save~~ saves it into his

C9 cont

computer. Since substitute data was used to replace protected pixel data, the user is unable to copy unmodified pixel data from the protected image.

Page 36, paragraph 5:

C10

Reference is now made to Figure 9, which is a simplified illustration of a system for copyright protection of digital images residing on a computer that are referenced in a web page residing on a different computer. Client computer 106 contains a web browser 112, which issues an HTTP request for a web page from server computer 900. The requested web page, 902, resides on server computer 900 (server computer #1), but it references a protected image 904 that reside on a different server computer 906 (server computer #2). As a result, server computer 900 may not be able to generate substitute data, such as encrypted image data, for protected image ~~104~~ 904 until it first downloads protected image 904.

Page 41, paragraph 5:

C11

The toolbar at the top of the screen indicates that the leftmost column, "Get List", is selected. A description of the toolbar is provided hereinbelow with reference to Figure 14. The file name "index.html" of an HTML page that is in the folder /Sample/csafe is highlighted in the left panel of Figure 13. The image files referenced within index.html are displayed in the right panel. As shown, they are files for GIF images. The "Status" column within the right panel indicates that none of the images listed in the panel are protected, since no protection icons appear. The protection management tool enables the user to select one or more of the listed images ~~listed~~ for setting protection. The user selects one or more images by clicking on their files names with the mouse, and using the "Shift" and "Control" keys to select a contiguous group of names or multiple names, respectively, as is the well-known standard for Windows operating systems. After selecting one or more images, the user clicks on the "Protect" button to have protection settings applied thereto.

Page 43, paragraph 1:

C12

The Tags button can be used when a user selects one or more HTML page file names, to protect images referenced within protection tags in the selected HTML pages. As described hereinabove, tags such as <!protect> and <!/protect> are used to delineate one or more sections within an HTML page, and the images referenced within the tagged sections can be protected by selecting the HTML file name and clicking on the "Tags" button. In distinction to the

C12 cont

~~Protect~~ "Protect" button which serves to protect all of the images within selected HTML pages, the "Tags" button only protects images ~~references~~ referenced within the tagged sections of selected HTML pages.

Page 43, paragraph 2:

The "Submit" button is used to confirm protection settings made by the user, and transmit them to the web server computer for application. When the user clicks on the "Submit" button, the protection settings he edited are sent to the web server computer and incorporated into the protection status database, as described hereinabove with reference to Figure 3 and Figure 4. Until the user clicks on the "Submit" button, the protection settings he edited are only displayed within the protection management tool by his local computer. Only when he clicks the ~~"Submit button"~~ "Submit" button are his settings actually applied. If the user does not click on the ~~"Submit "button~~ "Submit" button, then all of the protection setting he edited will not take effect, and the protection settings will remain at their former state if he closes the screen.

Page 44, paragraph 3:

C13

The topmost parameter is the IP address for the web server. The parameter setting indicated in Figure 15 specifies an IP address of 192.168.1.39 and a port of 80. The second parameter is the root directory for the web server, relative to which folder names and file names are specified. The parameter setting indicated in Figure 15 specifies a root directory of ~~d:/netpub/wwwroot~~ d:/Inetpub/wwwroot. The third parameter is the file name of a default web page that is displayed when a client first connects to the web server. The parameter setting indicated in Figure 15 specifies a default web page default.htm (residing in the root directory).

Page 45, paragraph 2:

C14

The sixth parameter indicates the image of a watermark to be used ~~to~~ for watermarking protected images, when the client is using an unsupported browser and when the third option above is chosen for handling unsupported browsers. Typically, the watermark image is a small image, and it is tiled so that the watermark appears repetitively in a checkerboard fashion, or other such fashion, over a protected image that is watermarked. The parameter setting indicated in Figure 15 specifies that the watermark image is in a file named watermark.gif. The seventh parameter indicates the saturation, or opacity level, with which the watermark is to be composited over a protected image,

C14 cont

when the client is using an unsupported browser. A saturation of 0.0 is fully transparent, and a saturation of 1.0 is fully opaque. The parameter setting indicated in Figure 15 specifies a saturation level of 85%. Preferably, this is the default parameter setting. The eighth parameter indicates a transparent color for the watermark; i.e., a color to be treated as background and not changed by the watermark. This ensures that backgrounds of protected images are not watermarked. The parameter setting indicated in Figure 5 indicates a watermark transparent color of white (255). Preferably, this is the default parameter setting.

Page 46, paragraph 2:

C15

The twelfth parameter indicates the directory in which substitute data, such as encrypted images, are cached for efficient re-use upon subsequent requests for the same protected images. The parameter setting indicated in Figure 15 indicates the directory /cache (relative to the root directory ~~d:/netpub/wwwroot~~ d:/Inetpub/wwwroot). The thirteenth parameter indicates the length of time during which a file is maintained in the cache directory. The parameter setting indicated in Figure 15 indicates a duration of 1,440 minutes. After this duration a cached file is purged from the cache. The fourteenth parameter indicates the frequency with which the cache is monitored, to determine which files are to be purged from the cache. The parameter setting indicated in Figure 15 indicates a monitoring frequency of every 1,440 minutes.

Page 49, paragraph 2:

C16

The user can check a box to update mirrors automatically, and then any edits he makes to parameter settings for the site currently being accessed will automatically be submitted to the mirror sites whenever the user clicks on the "Submit" button in the tool bar illustrated in Figure 14, to submit his edits to the web server computer. This mode of automatic update results in protection settings being updated incrementally in mirror sites each time the user edits them in one of the sites. However, if one or more edits are not synchronized with mirror sites, the mirror sites will ~~lost~~ lose synchronization and will not regain synchronization as future edits are made, even if the future edits are proliferated to the mirror sites. This loss of synchronization can happen, for example, if one of the mirror sties is not operational at the time the user makes his edits to the protection settings or, for example, if a mirror site is removed from the list of mirror sites.

Page 57, paragraph 1:

C17

The system extension patch for CopyBits() is preferably a head patch; i.e., the patch is applied and then the conventional system CopyBits() is called. The system extension patch for ~~OpenPicture()preferably~~ OpenPicture() preferably calls the plugin to update rectangles of the instances, and to set a flag to indicate to the system extension that the patch for CopyBits() should be used. The system extension patch for CopyBits() uses the rectangles and erases them on screen, so that the conventional CopyBits() call does not gain access to unmodified protected images. The patch for CopyBits() sets a flag indicating that the plugin should re-draw the images.

Page 59, paragraph 4:

C18

For another example, the present invention can be integrated with transaction software so that protected images can be purchased on-line. Specifically, when a user positions a mouse pointer over a protected image and right clicks on the mouse, a transaction menu can be popped up with one or more selections for purchasing the protected image. Selecting an option to purchase the image can trigger e-commerce transaction software. Thus when a user ~~tried~~ tries to save the image using the standard "Save Image As ..." command, he is notified that the image is copyright protected and presented with an opportunity to purchase the image. Selections for purchasing the image can include purchasing one or more hardcopy prints of the image, purchasing apparel, such as clothing, containing the image, and purchasing an electronic version of the image.